

# Reconstructing facade semantic models using hierarchical topological graphs

Yuefeng Wang<sup>1</sup> | Hongchao Fan<sup>2</sup>  | Guoqing Zhou<sup>3</sup>

<sup>1</sup>State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan, China

<sup>2</sup>Department of Civil and Environmental Engineering, Norwegian University of Science and Technology, Trondheim, Norway

<sup>3</sup>Guangxi Key Laboratory for Geospatial Informatics and Geomatics, Guilin University of Technology, Guilin, China

## Correspondence

Hongchao Fan, Department of Civil and Environmental Engineering, Faculty of Engineering, Norwegian University of Science and Technology, Trondheim 7491, Norway.  
Email: hongchao.fan@ntnu.no

## Funding information

National Natural Science Foundation of China, Grant/Award Number: 41431179 and 41771484

## Abstract

Semantic information in 3D building models is of vital importance for various applications in terms of smart cities. To infer the semantic information and localize the components on building facades, this article proposes a novel approach to model facades with semantics by constructing hierarchical topological graphs. This method utilizes the topological characteristics of building facades. In the first-layer layout graph, the algorithm takes the nearest cluster as the vertex and the distance between components as the edge. Thus, a topology graph is generated for the facade. The proposed algorithm is divided into three steps. First, the topology graph is obtained by calculating the spacing between the components. It is reasonable to calculate the topological graph by encoding the topological edges. If this calculation is not effective, the topology is justified by adjusting the spacing between components. Finally, the vertices in the graph are used to repair the occluded parts of the facade. In the second-layer graph, a grid is constructed according to the first-layer graph. Then, the attributes of the nodes are used to reconstruct the facade. The experimental results show that this method has a high accuracy of 90% and that the average time consumption is 6 s.

This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

© 2020 The Authors. Transactions in GIS published by John Wiley & Sons Ltd

## 1 | INTRODUCTION

The building facade is an important component of a 3D building in a digital city. Semantic information, such as the windows, balconies, and doors, can be used to analyze the building's structure and augment reality (Müller, Wonka, Haegler, Ulmer, & Van Gool, 2006). However, in previous methods, such as texture mapping, only the appearance of the facade is visualized; the semantic information of the facade is not actually established (Haala, Rothermel, & Cavegn, 2015; Zhou et al., 2016). Moreover, the texture image used for the facade takes the form of raster data. Therefore, large-scale 3D reconstruction is very time-consuming due to the limitations of a computer's graphics processing ability. In fact, computers can more efficiently process vector data than load raster-type texture images (Pu & Vosselman, 2009). In 2008, the Open Geospatial Consortium (OGC) passed the international standard CityGML (Gröger & Plümer, 2012), which uses a standardized model to represent the geometrical, topological, and semantic aspects of 3D building models (Uden & Zipf, 2013). Although this method provides a reliable solution for reconstructing a digital city, it also requires very tedious work to manually organize the building's attribute model. To automate this process, researchers propose recognizing and segmenting the semantics of facades from images or point clouds (Berg, Grabler, & Malik, 2007; Cohen, Schwing, & Pollefeys, 2014; Dai, Prasad, Schmitt, & Van Gool, 2012; Martinović, Mathias, Weissenberg, & Gool, 2012).

There are currently two main types of facade parsing methods: the data-driven method and the model-driven method. The former method extracts the image features and uses a machine learning framework to derive a classification model (Datta, Joshi, Jia, & Wang, 2008). The segmentation accuracy of this method is often affected by the image quality and the complexity of the facade. In addition, the process of training is less transferable. The latter methods use the synthesis of templates or grammatical rules to parse the facade (Müller, Gang, Wonka, & Gool, 2007). However, the setting of the template and the formulation of the facade grammar are subjectively determined by the designer. As a result, errors often occur when one uses a single grammar to handle different styles and different sizes for the facade.

To achieve the automatic reconstruction of various types of facades, Bao, Schwarz, and Wonka (2013) and Shen, Fan, Mao, and Wang (2016) explored the structural characteristics of facades. Because symmetry and alignment are the most common facade structures, they are also used to reconstruct facades. Indeed, these methods have good adaptability when reconstructing a symmetrical facade. However, this type of method cannot correctly reconstruct asymmetric facades. In addition, the present methods lack reliable theoretical support.

To explore a more robust method of facade reconstruction, this article proposes a hierarchical topological graph to reconstruct a semantic facade model. First, we explore the overall layout of the facade and propose the first-layer layout of the topological graph to deduce the facade's layout. Based on the field of architecture, a constraint function is developed to judge the rationality of the layout. The second-layer graph is then used to reconstruct the facade. According to the deduced facade layout, we initially construct a grid in which each node acts as a vertex in the graph. By judging the attributes of the vertices, we traverse the entire grid to achieve facade reconstruction.

The main contributions of our method are as follows:

1. Using a topological graph to describe a facade is a clear, intuitive, and lightweight approach that is helpful for the rapid reconstruction of large-scale 3D urban models.
2. The proposed hierarchical topological graph is a flexible approach. This approach can not only verify the reasonable layout of a facade, but also search for a proper layout for occlusion and deletion facades.
3. Topological correlations on the facades are explained using the principle of architectural form. On this basis, the geometric constraints of the facade layout are proposed, and the semantic entities are reconstructed.

The remainder of the article is structured as follows: we analyze the previous methods regarding facade parsing and facade reconstruction in Section 2. According to the analysis, we have identified two main shortcomings

of the previous methods. To address these shortcomings, we propose a hierarchical graph to parse and reconstruct the facade. First, we introduce the layout control function and its theoretical support (i.e., the principle of the architectural form) in Section 3. Next, we introduce the proposed hierarchical topological graph in Section 4. Then, we set the unique meaning for the vertices and edges. An illogical graph that can be rectified according to the algorithm for searching graphs is introduced in Section 4 as well. Some experiments for different facades are reported in Section 5. Finally, conclusions and suggestions for future work are given in Section 6.

## 2 | RELATED WORK

### 2.1 | Data-driven methods

This type of method mainly studies the visual features of facade images. The underlying features—such as the color, gradient, texture, and shape—can be obtained easily by computer vision theory. By learning the features of interesting semantic entities, we can derive a unique equation to describe them. Using the feature equation to build a classification model is a simple way to achieve semantic segmentation of facade images.

In the past 20 years, researchers have analyzed images by summarizing the shapes, edges, gradients, and color features of objects (Datta et al., 2008). Among these features, the edge is often used to extract an object on the facade because manually constructed buildings always have legible edges. Lee and Nevatia (2004) judged the arrangement of windows by counting the cumulative values of the image edges in the horizontal and vertical directions. This method assumes that windows are rectangular. However, the objects on a facade are various. To extract the edges of facade components more accurately, some researchers have semi-automatically combined the line features and inferred the structures of the windows according to the context (Wenzel & Förstner, 2012; Xiao et al., 2008). As a result, the robustness of window extraction has been improved. In some facade images, the changes in light and scene cause the edge information of the facade components to not be completely obtained. Lasers can detect the edge information on the facade sensitively, and the process is not affected by ambient light. Therefore, some researchers have proposed extracting the facade edge with the help of a laser point cloud. For example, Wang, Ma, Zhu, Zhao, and Liao (2018) took advantage of the fact that windows can form holes in point clouds to detect windows. This method enhances the edge constraint of windows. However, laser point clouds are usually not easy to obtain.

Previous studies have shown that it is not easy to extract windows from images only using edge features. To obtain a better adaptive facade segmentation model, many researchers have gathered an enormous number of images and established image datasets to train advanced features (Riemenschneider et al., 2012; Teboul, Kokkinos, Simon, Koutsourakis, & Paragios, 2011; Tyleček, 2012). Through the annotated ground-truth images, researchers can train the right samples and wrong samples by means of machine learning. Gadde, Jampani, Marlet, and Gehler (2018) used a conditional random field (CRF) to train facade images. Apart from considering the image features (e.g., location of pixels, RGB color space, or histograms of the oriented gradients), this method also adds the contextual features of pixels. The advantage of this method is that it can be applied to different facade images, but the accuracy of window extraction is not high. Teboul, Simon, Koutsourakis, and Paragios (2013) segmented facades using random forests. In this method, it is impossible to accurately judge the types of facade components. Therefore, only typical Haussmannian-style Paris architecture can be correctly segmented, because typical Paris architecture has a unified type. Cohen et al. (2014) employed dynamic programming to constrain the size and position of each object in a probability graph. This method improves the accuracy of the facade segmentation, but it cannot be applied to many types of facade. Jampani, Gadde, and Gehler (2015) used context information to enhance decision trees and optimize the results of facade segmentation. Fathalla and Vogiatzis (2016) adopted a restricted Boltzmann machine to globally optimize the probability graph. This method implements segmentation at the pixel level and adds steps for the global optimization of each pixel. In addition to these referenced machine learning methods, in previous research support vector machines (SVMs) and simulated annealing (SA) have also

been used to segment facade images (Datta et al., 2008). In recent years, the theory and method of deep learning has been developed rapidly, and the underlying features have been abandoned due to the low robustness. According to the principle of deep learning, very rich image features are obtained by the process of convolution. Moreover, a deep network structure also enhances the robustness of semantic segmentation (Lotte, Haala, Karpina, Aragão, & Shimabukuro, 2018; Schmitz & Mayer, 2016; Zhang & Liang, 2017).

Although feature-based image segmentation has achieved great success, these methods cannot solve the problems of occlusions, shadows, and deletions. To meet these challenges, people have begun to pay attention to the content of images. Cheng, Zhang, Mitra, Huang, and Hu (2011) proposed the method of region contrast (RC). RC depends on an interesting object being identifiable by high contrast with the surrounding pixels. This high contrast can be extracted according to the spatial relationship, which can also be described as the geometric context. Hu, Zhang, Wang, Martin, and Wang (2013) established a hierarchical graph to express the structures of images, where the edge in a graph denotes the geometric context. Mathias, Martinović, and Van Gool (2016) applied context information of pixels to optimize probability graphs. Moreover, the authors also optimized the segmentation results by applying their prior knowledge of buildings. Both the prior knowledge and the geometric context need to be adjusted according to the scene of a facade. However, these methods hardly detect noise and automatically reconstruct facades. There are some rich facade types that require very substantial prior knowledge in order to optimize the segmentation results.

## 2.2 | Model-driven methods

In contrast to data-driven approaches, model-driven approaches use combinatorial models to represent objects. By setting the model parameters, a facade can be segmented and reconstructed in the form of a parameter set. A set of suitable models can also be used to repair the occlusions and deletions on a facade according to a prediction model. Facade grammar is the most common model for reconstructing facades.

Alegre and Dellaert (2004) first proposed the splitting grammar to parse facades. The splitting grammar is a type of context-free grammar that can be denoted as  $G = \langle N, T, S, P \rangle$ , where  $N$  is a finite set. Each element  $n \in N$  is called a *non-terminal character*. This character indicates that the process of segmentation cannot be stopped. It contains separable parts, such as floors, roofs, etc. These non-terminal characters can be replaced by other non-terminal characters or terminal characters.  $T$  is the finite set of terminals, and it contains the types of semantic entities—such as windows, balconies, and doors. When a terminal character appears, the splitting process is over. In addition,  $S$  is the starting symbol and  $P$  is the set of production rules.  $P$  has two common rules (i.e., vertical splitting and horizontal splitting). The limitation of this method is that the splitting grammar cannot automatically adjust different types of facade. Koutsourakis, Simon, Teboul, Tziritas, and Paragios (2009) tried to explore a more general grammar. They created a set of generic rules for a Haussmannian building.<sup>1</sup> Then, by means of reversible-jump Markov chain Monte Carlo (RJMCMC), these authors assigned a probability to each generic rule. Therefore, for different facades, the synthesis of the rules can achieve an optimal solution. Tyleček and Šára (2010) also used RJMCMC to establish a grammar for a simple facade. Gadde, Marlet, and Paragios (2016) summarized several types of grammars and simplified them. As a result, a more general generic grammar was designed. Using reinforcement learning, these authors applied the grammar to handle different styles of facades. Although these proposed grammars can be combined in many forms, they cannot handle irregular facades.

In addition to the above two methods, hybrid-driven methods are also being explored. Becker (2009) combined image segmentation with grammar to reconstruct facades for the first time. Mathias et al. (2016) used a framework of three layers to parse facades. The first layer employed a recursive neural network (RNN) to obtain a probabilistic interpretation of each pixel. The second layer merged the specialized facade components using a Markov random field. In the third layer, the weak architecture principle was used to infer the procedural shape grammar. The three-layer structure improves the accuracy of facade parsing.

In all of the above image segmentation and reconstruction methods, an enormous dataset is needed to train the segmentation model or grammar. Moreover, due to the types, sizes, locations, distribution rules, and types of facade components varying, the proposed methods cannot have viable mobility. Therefore, these methods are unable to process large quantities of facade image data. The disadvantages of the previous methods can be summarized in two points as follows:

1. Prior methods weakly cope with images that have occlusions, shadows, and deletions.
2. The established grammars are always focused on the arrangement of elements on the facade, and a long sequence is usually required to represent the facade. Moreover, the search space for a reconstruction is enormous.

### 3 | TOPOLOGICAL CONSTRAINTS OF THE FACADE LAYOUT

#### 3.1 | Principle of the architectural form

Incorrect semantic segmentation often results in the confusion of facade components. For example, the results of facade segmentation usually produce very large or very small windows. Moreover, the locations of the semantic entities are often not corrected. Although rules-based constraint approaches have been applied to optimize the results of segmentation, all of these methods lack theoretical support. Therefore, we first introduce the principle of architectural form.

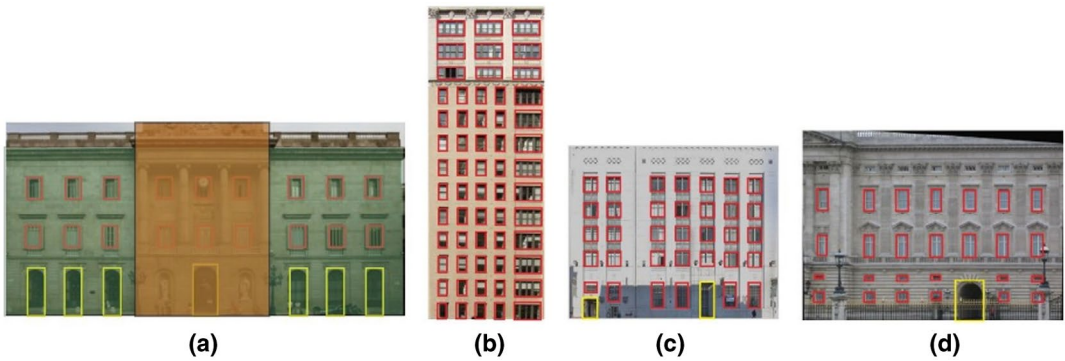
The composition of architectural elements, the exact permutations of a building mass, and the proportion of one part to the other parts are crucial factors in the design of an architectural form (Doersch, Singh, Gupta, Sivic, & Efros, 2012; Flemming, 1990; Jennath & Nidhish, 2016). According to the principle of architectural form, there are five types of constraints.

##### 3.1.1 | Varying and uniform

In most facade compositions, differences exist in their layout. These differences reflect the degree of importance of local layout and the elements' composition (Ching, 2014). For an important part of a facade, the emphasis must be achieved with a form of exceptional size, unique shape, or strategic location. The primary and secondary portions of a facade can be separated into two categories: (a) one primary and two auxiliaries; (b) one primary and one auxiliary. For the first category this mainly refers to a symmetrical layout whose central parts are always the primary one and where wings can be regarded as secondary. Its common form is shown in Figure 1a. Another category usually shows its primary portion on one side of a facade rather than in the center, as in Figure 1c.

##### 3.1.2 | Contrast and harmony

Various semantic entities, such as doors and windows, can cause confusion when they have close size and arrangement. Because when the difference is lost, nothing is emphasized (Ching, 2014). Contrast means increasing the difference between elements. For example, on the ground floor, if a door is regarded as the primary part, it must be larger than the windows. Taking public buildings and hotels as an example, their doors are usually significantly larger than other facade objects, and have pillars and other decorations to highlight their importance. Contrast should also be concerned with multiple local layouts of a facade. The center of a symmetrical facade needs more elements than wings to highlight its characteristics (Krier & Vorreiter, 1988). It is noteworthy that a strong contrast



**FIGURE 1** Examples of satisfying the topological constraints: (a) classical symmetrical facade with one primary and two auxiliaries; (b) although there are two sizes of windows on this facade, they all have the proper proportions to create a proper scale between the windows and the façade; (c) facade designed as an asymmetric form of a primary part and an auxiliary part; and (d) the windows at the bottom make the entire facade look more in equilibrium

usually does not appear on facades, because buildings typically have stable structures. Furthermore, the elements also need to have orderly arrangement and consistent style, for instance windows on the same floor are normally aligned center to give people a sense of order.

### 3.1.3 | Proportion and scale

A proper proportion does not only ensure the stability of the facade structure, but also reflects the sense of order in a visual structure. The proper proportion has been determined by people according to long-term practice (Stamps, 1999). For example, for a single French door, its height is usually greater than its width, based on the proportion of the human body. The proportion of the total area of windows to the size of the facade also has a reference standard (European Parliament, 2018), which can help to calculate the daylighting area. In the dimension design of facade elements, in addition to a proper proportion, we should also pay attention to the scale. Scale refers to the size of an element compared to a reference standard or the size of others. For instance, the relationship between the floor height and the door height can be used to determine the appropriate size of the facade elements in an image.

### 3.1.4 | Symmetry and equilibrium

With the force of gravity, a state of equilibrium can be expressed by situating the centroid of the facade on a central axis, with the position of the centroid below the midpoint. This is subject to human cognition. A symmetrical condition is always expedient in constructing a state of equilibrium (Salvan & Thapa, 2000). The symmetry can be utilized to organize a facade layout in two ways: (a) global symmetry; and (b) local symmetry. The former must maintain strict similarity on opposite sides of a median axis, as shown in Figure 1a. At some point, however, the functional requirements or environmental constraints of a facade layout restrict the global symmetrical condition. Therefore, the latter case of local symmetry can serve for complex situations. In this case, the size and position of each local layout should be designed to achieve a state of equilibrium, such as in Figure 1c.

### 3.1.5 | Cadence and rhythm

Almost all facades incorporate elements that are, by their nature, repetitive. In order to satisfy functional and aesthetic demands, architects use the 'rhythm' to organize the facade elements (Salvan & Thapa, 2000). Rhythmic patterns can be emphasized by geometric features and the placement of facade elements. In practice, elements with a common trait can be organized at regular intervals, and different types of elements appear on the facade alternately. This principle can provide continuity and lead us to deduce a complete facade layout from a defective image. Moreover, any break in rhythmic pattern should verify the importance of the interrupting element or interval.

## 3.2 | Control function

It must first be stated that there are some special types of buildings, such as all-glass curtain facades and twisted facades. It is difficult to find a fixed constraint to describe the layout of these facades. Therefore, we only consider the most common facades in this work. We believe that the most common facades are in accordance with the principle of architectural form. The following control functions are designed by inspecting a large number of facade structures.

$R_1(n)$  represents the ratio of the sum of the component widths to the facade widths. The equation is used to control the components to be not too crowded or sparse on one floor. When there are only windows on a floor, we calculate the ratio of the width of the windows to the width of the facade. When there are not only windows but also doors on a floor, we calculate the ratio of the total width of the windows and the doors to the width of the facade. The equation  $n/(2n - 1)$  in  $R_1(n)$  is the best value of the error-tolerant rate that we can sum up by observing different styles of facades.  $R_2(n)$  is designed by utilizing the daylight efficiency in energy-saving designs of building facades. The equation can be used to constrain the area and number of windows on the facade (Li & Lam, 2001). When we determine the size of the window, the number of windows  $n$  can reflect the transmittance of the facade.  $R_3(n)$  is a combined constraint of  $R_1(n)$  and  $R_2(n)$ . Not only do we need to determine the best window distribution on one floor, but we also need to calculate the optimal layout of the entire facade. We combine  $R_1(n)$  and  $R_2(n)$  to calculate the optimal facade layout under two control functions. The equation can be used to evaluate the inferred facade layout as follows:

$$R_1(n) = \left| \frac{\sum_{i=0}^n w_{components}(i)}{W_{facade}} - \frac{n}{2n-1} \right| \quad (1)$$

$$R_2(n) = \left| \frac{\sum_{i=0}^n w_{windows}(i) \cdot h_{windows}(i)}{W_{facade} \cdot H_{facade}} - 0.3 \right| \quad (2)$$

$$R_3(n) = R_1(n) + R_2(n) \quad (3)$$

where  $w_{components}$  and  $h_{components}$  are the width and height of the facade components, respectively. We use three types of components in this work: components = {window, balcony, door}.  $W_{facade}$  and  $H_{facade}$  are the size of the facade, and  $n$  is the number of components. The value of 0.3 in  $R_2(n)$  represents the regular ratio of the window to the entire facade. This value was obtained from the statistics of many building codes (Li & Lam, 2001).

When we reconstruct the facade, the width and the number of elements can be adjusted appropriately to satisfy the control function. In our experiment, we must find a proper layout which can guarantee that the value of  $R_3(n)$  is minimal—that is,  $n = \arg \min (R_3(n))$ .

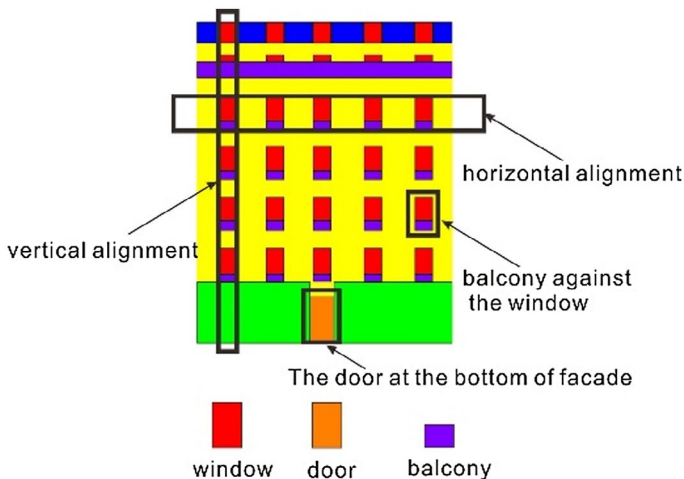
## 4 | HIERARCHICAL TOPOLOGICAL GRAPHS

In many tasks, topographical graphs have been used to describe complex geometric objects by employing the relationship between vertices (Felzenszwalb & Huttenlocher, 2004; Hsu, 2004; Ladicky, Russell, Kohli, & Torr, 2009; Yang & Förstner, 2011). In this work, we propose hierarchical topological graphs for reconstructing a facade. The first layer of the graph is inspired by the principle of architectural form. On the facade, the topological properties can easily be verified, as shown in Figure 2. For example, the arrangement of the windows is vertical- and horizontal-aligning, and the balcony is always under the window. The proposed method presents the facade as an overall layout by verifying the rationality of the facade, as described in Section 4.1.

The second layer of the graph is designed to reconstruct the facades. We construct a grid with the deduced facade layout (Figure 3), where the nodes in the grid act as vertices in the graph. We set two types of vertices, such as the existing components and the undetermined vertices. For the undetermined vertices, there are two categories of attributions, namely “components” and “empty.” We cope with the undetermined vertices by obtaining the attribution of the adjacent vertices. Two constraints have been drawn up to accelerate the process of traversing, as described in Section 4.2.

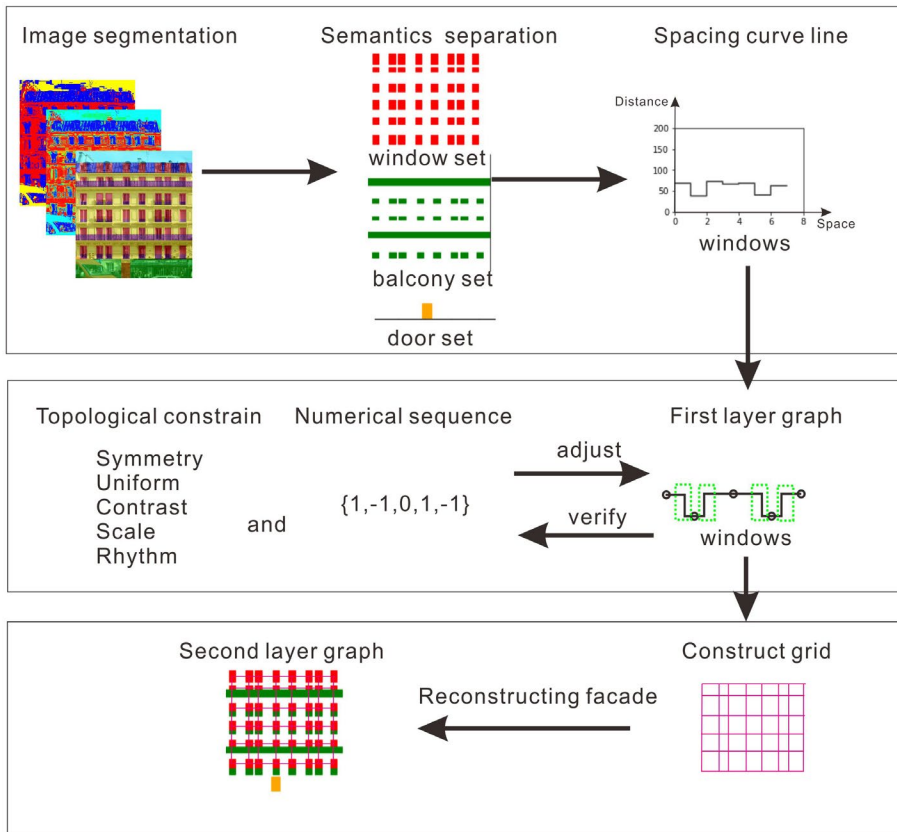
### 4.1 | First-layer layout graph

In the first-layer graph, we assign specific geometric meanings to the vertices and edges. We set  $G_{facade} = \langle V, E \rangle$  as an undirected graph, where  $v \in V$  represents vertices that are composed of the nearest cluster. The cluster is a clique that has at least one element, which can be determined by the principle of “Gestalt Laws: Laws of Proximity.” A 4-tuple (type, number, height, width) denotes the attribution of the vertices and contains the semantic information on the components, the number of components in the cluster, and the size of a single component. Two adjacent vertices are connected by edges—that is,  $e(v_i, v_{i+1}) \in E$ . An attribution  $e(v_i, v_{i+1}) = (d_{low}, d_{high})$  needs to



**FIGURE 2** Topological properties of a facade





**FIGURE 3** Flowchart of the hierarchical topological graph

be attached to an edge, where  $d_{low}$  means the intra-cluster distance and  $d_{high}$  means the inter-cluster distance; the relationship between these features is shown in Figure 4. When there is only one element in a cluster,  $d_{low}$  can represent the inter-cluster distance of the previous edge.

In particular, when there are two different edges between two vertices, we add a “null” vertex between the two edges so the form of the graph remains intact, as described by  $e_2$  in Figure 4. When there is only one vertex in a graph, such as the uniformly distributed layout, the edge can be represented as  $e(v, \text{“null”}) = (d_{low}, 0)$ .

In theory, the method of constructing a first-layer graph is the same as the principle of “Gestalt Laws: Laws of Proximity.” This straightforward law states that items close to each other tend to be grouped together, whereas items further apart are less likely to be grouped together (Schwartz & Krantz, 2017).

### 4.1.1 | The types and attributes of edges

In a first-layer graph, we set four types of primary edges, as shown in Figures 5a–c, and f. All of these types can be determined according to the spacing between the components. We will explain this in its simplest form. Suppose there are three windows on the same floor, and the spacing distances between two adjacent windows are different. The two adjacent windows with smaller spacing distance would form a cluster. The smaller spacing distance is then the intra-cluster spacing. The remaining window then forms a cluster of a single element, and the larger spacing distance is the inter-cluster spacing. As a result, the edges between the two clusters are formed. For example,

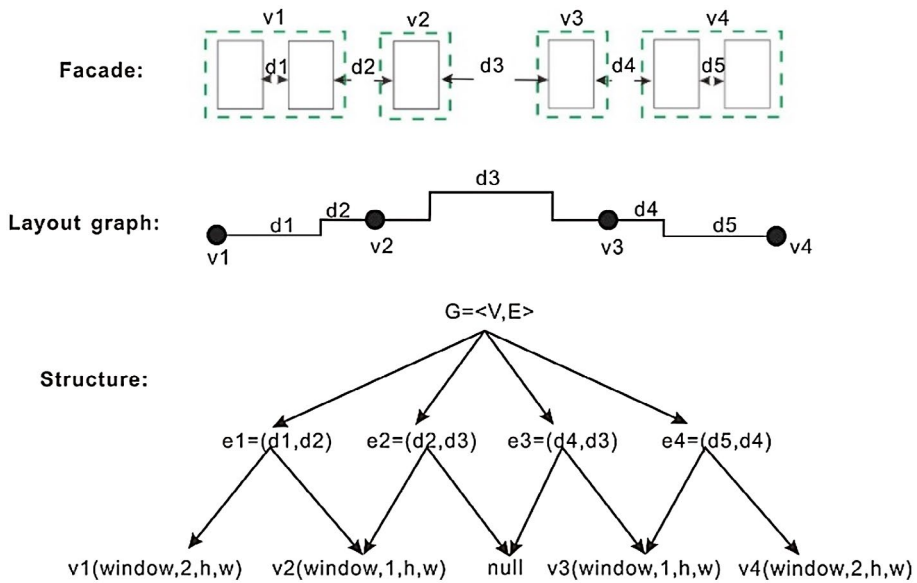


FIGURE 4 Topological graph and its structure

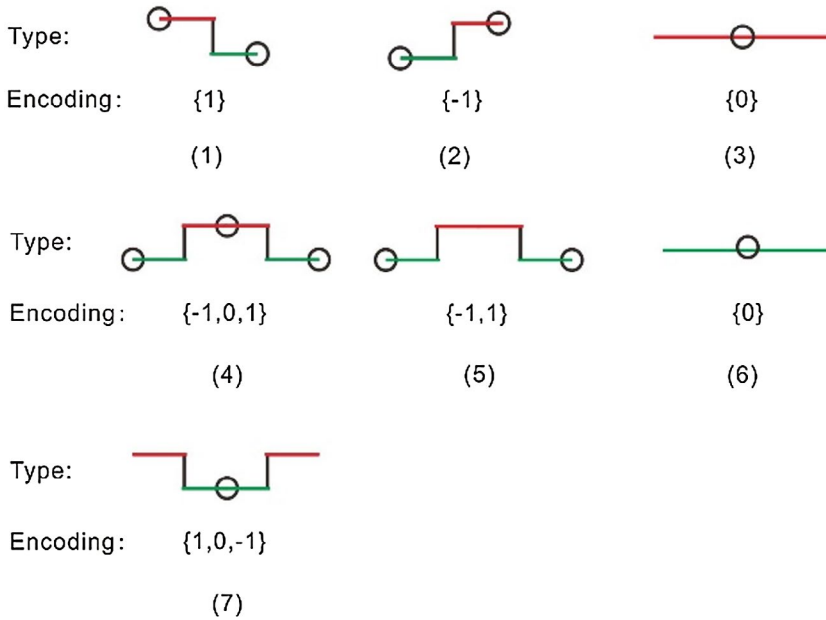


FIGURE 5 In this type of polyline, the little rings represent the vertices and the broken lines represent the edges in the graph

In Figure 5a we present the inter-cluster spacing as red lines and the intra-cluster spacing as green lines. The edge in Figure 5a indicates that the distance between the two vertices decreases from left to right. The other edges can also be explained according to their shape.

In addition to setting the types of edges, we also assign different numbers to different types of primary edges, since we need to explore a more flexible layout-deducing algorithm. In previous structure-based methods, simply relying on symmetric conditions to test the rationality of the layout would result in a monotonous facade.

Moreover, different sizes and distribution types of the components also need to be considered in a complex facade. In the proposed method, we believe that a facade layout can be described as a combination of different types of edges. It is also possible to verify the rationality of a layout by numbering the edges as in Section 4.1.3. For example, we set the direction of Figure 5a to be positive and identify it as “1.” Conversely, the direction of Figure 5b is negative and identified as “-1.” When a facade is symmetrical, it is easy to judge that the sum of the numbers on the facade is 0. This is because, for a symmetrical facade, the directions of the edges must be opposite to one another. The edge of the straight line in Figure 5c happens when the intra-cluster spacing is equal to the inter-cluster spacing, and this situation is identified as “0.” The combination of edges in Figures 5d and e is a contrasting example that demonstrates the existence of “0.” In fact, the most representative situation is shown in Figure 5g. When the graph is concave, there is always a cluster in the valley, such as in Figure 5f. The reason is that the intra-cluster spacing must be less than the inter-cluster spacing on one edge, according to the principle of “Gestalt Laws: Laws of Proximity.”

### 4.1.2 | The meaning of the vertices

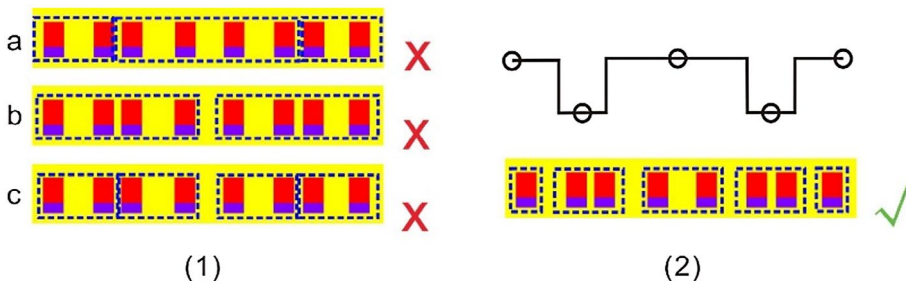
In the defined graph, a vertex is a nearest-neighbor cluster composed of one or more elements. Because the attributes of the edges also include intra-cluster spacing and inter-cluster spacing, we have no way of determining the appropriate composition of the vertices in the process of adjusting the graph, as shown later in Figure 9. When the layout graph is determined, the composition of the vertices can be determined. According to the principle that the inter-cluster spacing is less than the intra-cluster spacing, we can easily determine the composition of the vertices. A valid example is shown in Figure 6b.

We can also use topography graphs to identify some representative facade structures as in Figure 7. The topological graphs we used are combinations of simple edges. We also describe these graphs in the form of numerical sequences. One example of the regular type can be expressed as {0}, two examples of the concave type can be expressed as {1, 0, -1} and {1, -1, 0, 1, -1}, and three examples of the convex type can be expressed as {-1, -1, 1, 1}, {0, -1, 1, 0}, and {-1, 0, 1}.

### 4.1.3 | The algorithm to search for a proper layout graph

We use the control function and numerical sequence to design an algorithm in order to search for the proper layout. The algorithm can verify whether the facade layout is reasonable. For occluded and defective facades, we can also use this algorithm to deduce a rational layout.

The derivation process of the layout is shown in Figure 8 and is divided into three steps: (a) calculating the initial layout; (b) verifying the layout; and (c) adjusting the layout.



**FIGURE 6** Determination of the clusters

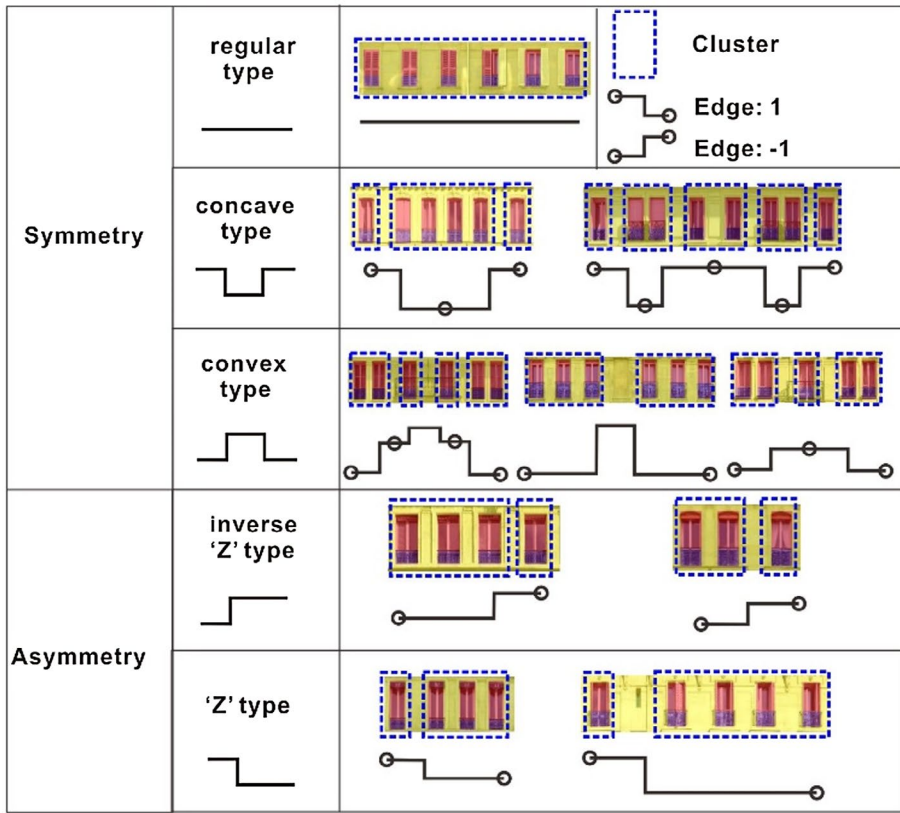


FIGURE 7 Complex combination of topographic graphs for the representative facade

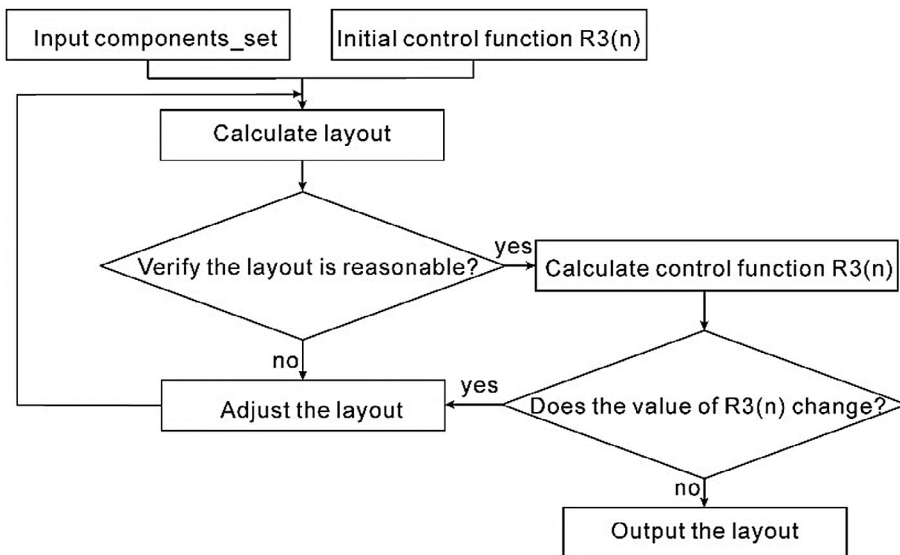


FIGURE 8 The flowchart of the searching layout

(1) *Calculating the initial layout.* By using the position and number of the facade components, we can calculate the spacing distance between the adjacent components and derive an array  $D = \{d_0, d_1, \dots, d_i, \dots, d_{n-i}, \dots, d_n\}$ . Then, we can construct the initial layout graph  $G1 = \langle V, E \rangle$  and the corresponding numerical sequence  $S = \{s_0, s_1, \dots, s_m\}$  by using,

$$s_i = \begin{cases} -1 & d_{i+1} - d_i > t \\ 0 & d_{i+1} - d_i < t \\ 1 & d_i - d_{i+1} > t \end{cases} \tag{4}$$

where  $i \in (0, n)$ ,  $n \in \mathbb{N}^*$ . Because some errors always occur in image segmentation (Lotte et al., 2018), we have designed a threshold value  $t = \min(w_{components})$  to tolerate the error, where  $w_{components}$  means the width of components. In addition, multiple zeros can be considered as one vertex in a sequence. Therefore, they need to be merged. There is one exception, as shown in Figures 5d and e.

(2) *Verifying the layout.* By analyzing the numerical sequence of the layout, we believe that the following conditions should be satisfied for a rational layout:

- If we start from the middle and step to both ends in a numerical sequence, the sum of the symmetrical position must be equal to zero.
- For asymmetric facades, there is only one non-zero value.

These conditions are strong constraints because they fit most common facades. According to the above conclusions, the rationality can be determined by analyzing the numerical sequence  $S$ . If the initial layout does not conform to the above conditions, the layout is adjusted according to step 3 below. If the condition is satisfied, the value of the control function is calculated. To obtain the optimal facade layout, we need to adjust the layout several times and calculate the control function. When the value of the control function no longer changes, we continue the algorithm flow. At this time, we select the layout with the smallest value in Equation (3) as the optimal layout.

(3) *Adjusting the layout.* First, the values in the symmetrical positions of array  $D$  are compared. If the difference between two elements is greater than the sum of the average window width and the smaller values, the larger values are split, as in the following:

$$Diff = |d_i - d_{n-i}| \quad (i = 1, 2, \dots, n) \quad (n = \text{length of } D) \tag{5}$$

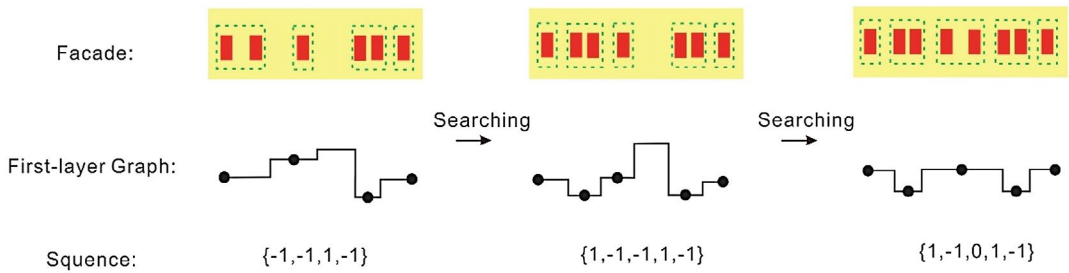
If  $Diff > w_{components} + \min(d_i, d_{n-i})$ :

$$j = \arg \max(d_i, d_{n-i}) \tag{6}$$

$$(d_j, d_{j+1}) = \begin{cases} (d_{n-i}, d_i - d_{n-i} - w_{components}) & \text{when } j = i \\ (d_{n-i} - d_i - w_{components}, d_i) & \text{when } j = n - i \end{cases} \tag{7}$$

Thus, we obtain a new array  $D_{new} = \{d_0, d_1, \dots, d_j, d_{j+1}, \dots, d_{n-i}, \dots, d_n\}$ , when  $j = i$ . We then advance the new array into step 2 to restart the verification.

Figure 9 is an example of using a layout search algorithm to deduce the layout. According to the definition of the layout sequence, we obtain an initial facade layout, as shown in the left-hand graph of Figure 9. Because this



**FIGURE 9** Process of the layout search algorithm

layout does not conform to our conditions of a rational facade layout, the search procedure needs to be repeated. After searching, the proper layout is shown on the right-hand side of Figure 9.

#### 4.1.4 | Layout reconstruction

From Section 4.1.3 we can infer the facade layout represented by the graph. We designed a pseudo-code to represent how to use the graph  $G_{facade} = \langle V, E \rangle$  to reconstruct the layout of the facade:

---

##### Algorithm 1. Reconstruction Layout

---

Input:  $G_{facade} = \langle V, E \rangle$

Output: Layout

For  $i \leftarrow 1$  to  $length(E)$

$e1 \leftarrow E[i-1]$ ;  $e2 \leftarrow E[i]$

IF  $e1[1] \leq e2[0]$ :

For  $j \leftarrow V[i-1].num$

Layout.append ( $V[i-1][j].type$ ,  $V[i-1][j].width$ ,  
 $V[i-1][j].height$ )

Layout.append ( $e1[0]$ )

End For

Layout.append ( $e1[1]$ )

Else:

IF  $V[i] \neq null$

skip  $V[i]$

Else:

For  $j \leftarrow V[i-1].num$

Layout.append ( $V[i-1][j].type$ ,  
 $V[i-1][j].width$ ,  $V[i-1][j].height$ )

Layout.append ( $e2[0]$ )

End For

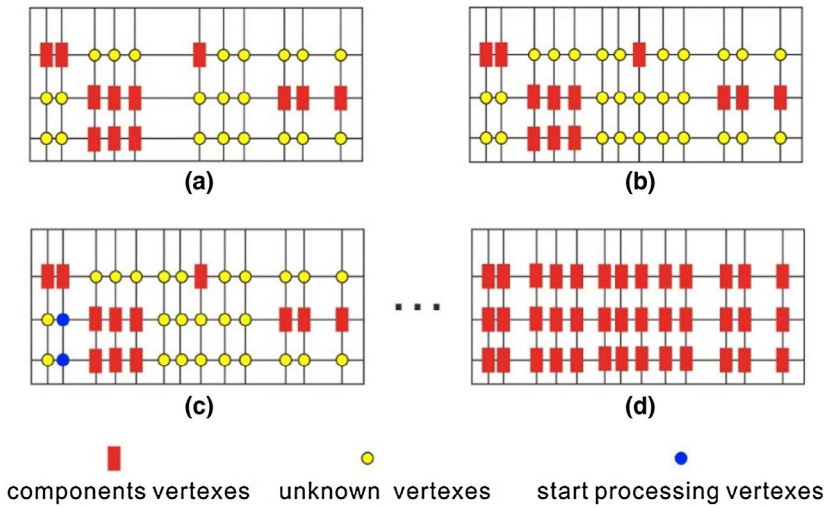
Layout.append ( $e2[1]$ )

End IF

End IF

End For

---



**FIGURE 10** Process for reconstructing the facade. Here, (a) is an initial facade, (b) is an integrated grid obtained according to the proper layout, and (c) and (d) are processes of assigning attributions to unknown vertices

### 4.2 | Second-layer graph

The second-layer graph is designed to reconstruct the facade. First, we construct the grid  $(m, n)$  by using the layout deduced from the first-layer layout graph.

Here,  $m$  is the number of rows in the grid, which can be obtained by calculating the locations of the known components with the equal difference series equation. In addition,  $n$  is the number of columns in the grid, which can be obtained according to the following equation:

$$n = \left( d_i + \frac{W_{components}}{2} \right) \quad (i = 1, 2, \dots, \text{length}(D) - 1) \tag{8}$$

In the second-layer graph, the nodes in each grid are regarded as vertices of the graph (Figure 10). The connection between the vertices and the surrounding nodes serves as the edge of a graph. The vertices are then classified into two categories, namely “known” and “unknown,” where “unknown” vertices need to be determined by the attributes of the adjacent edges. The attributes of the edges include the number and attributes of the adjacent vertices. Therefore, the second-layer graph can be denoted as  $G_2 = (V, E)$ , where  $V = (\text{position}, \text{attribution})$  and  $E = (\text{num}, \text{attributions})$ .

#### 4.2.1 | Reconstructing a facade using the second-layer graph

We traverse every unknown vertex and assign attributes to them by judging the attributes of the edge. Each unknown vertex has two possible attributes, namely “component” and “empty.” If the number of vertices is  $n$ , the computational complexity is  $2^n$ . Because this procedure is very time-consuming for complex facades, we use the topological constraints of the facade components to define two constraint conditions:

- *Constraint condition 1.* If there are four “empty” vertices around an unknown vertex, the vertex can be considered as a “component.”

- *Constraint condition 2.* If there are “component” vertices around an unknown vertex, the vertex is assumed to be a “component.” Next, it is determined whether the component intersects with the surrounding components. If there is an intersection, the unknown vertex is “empty;” otherwise it is identified as a “component.”

The reconstruction process is divided into two steps:

1. *Identifying the locations of the unknown vertices and the attributes of their edges.* We first sort the values in the attributions of all unknown vertices. We start with the unknown vertices that have the most known adjacent vertices. For the processed unknown vertices, we assign an attribute and mark them as a known vertex until the number of unknown vertices equals 0.
2. *Assigning attributions to unknown vertices.* For the retrieved unknown vertex, we first determine whether it satisfies *constraint condition 1*. If it satisfies this condition, it can be directly marked as empty. Otherwise, the attributes of the unknown vertices are determined according to *constraint condition 2*.

## 5 | EXPERIMENT AND DISCUSSIONS

The experimental data used in this article are building images obtained from the crowdsourced data-sharing website Flickr. These building images were taken and uploaded by volunteers through smartphones and smart cameras without professional guidance, and they are taken according to the preferences of the volunteers. Therefore, these images are stylistically diverse. Moreover, because volunteers have different shooting equipment and methods, the quality and resolution of the images also vary greatly. In this experiment, we randomly select building images without considering the area and equipment. In this progress, we also select images in the ECP2011 Haussmannian dataset, which has been used in other advanced methods. A method based on reinforcement learning of the shape grammar (Teboul et al., 2011) is used for comparison with the proposed graph-based method.

To determine whether the proposed method is acceptable compared to the advanced method, the ground-truth of the selected images has been constructed by searching the corresponding architectural images in Google Street View.

### 5.1 | Result

First, geometric adjustment is needed, which includes the calculation of the minimum bounding rectangle (MBR) and the position adjustment of the semantic entities (Liu, Zhang, Zhu, & Hoi, 2017). In this process, we do not limit the window to the same size because the types of windows are different and windows on the same floor may be different in size. The position adjustment contains two steps: (a) the balcony needs to be centered on a vertical line with adjacent windows; and (b) when the same type of elevation member is on the same floor, its geometric center must be on the horizontal line.

It is worth mentioning that in the facade reconstruction process we used different topologies to represent the layouts of different types of semantic entities.

#### 5.1.1 | Facades with regular structures

Symmetrical facades are the easiest to detect because their structures can easily be verified by the first-layer layout graph mentioned in Section 4. Therefore, all the facades in Figure 11 have accurate reconstruction results. Observe that windows are not constrained to a uniform size in the reconstructed image. As mentioned above, we have considered the differences in the window types and sizes. In fact, this issue also makes the proposed method susceptible to facade semantic segmentation.



### 5.1.2 | Reconstruction of irregular facades

We select several irregular facades to test the effect of reconstruction (Figure 12). For example, Bld-5 and Bld-6 are asymmetrical facades that can be derived from the proposed first-layer layout graph. As seen from the reconstructed image, the results are accurate. Bld-7 is a facade with alternating windows. This type of facade is difficult

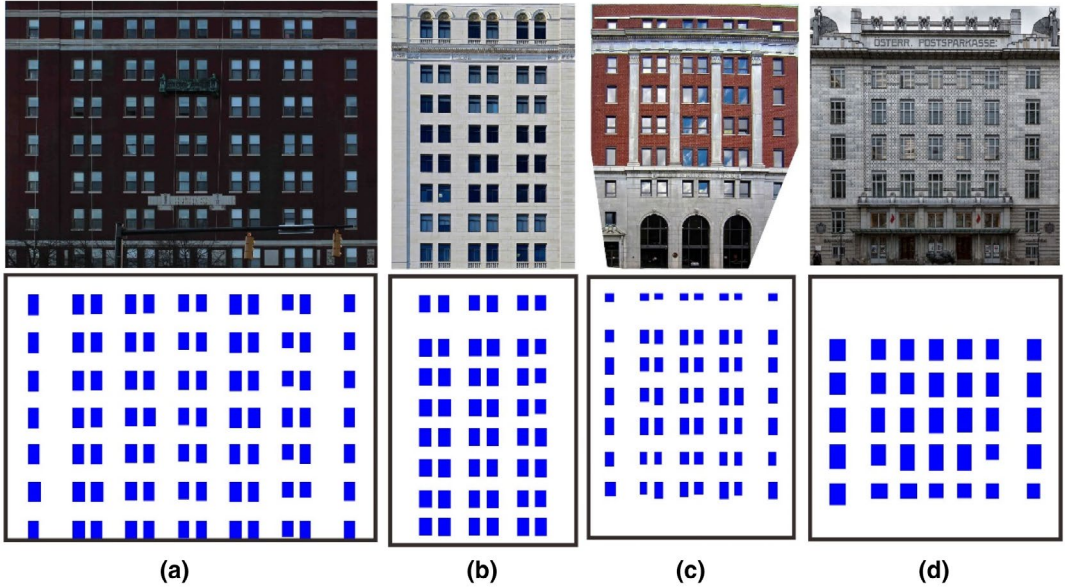


FIGURE 11 Facades with regular structures: (a) Bld-1; (b) Bld-2; (c) Bld-3; and (d) Bld-4

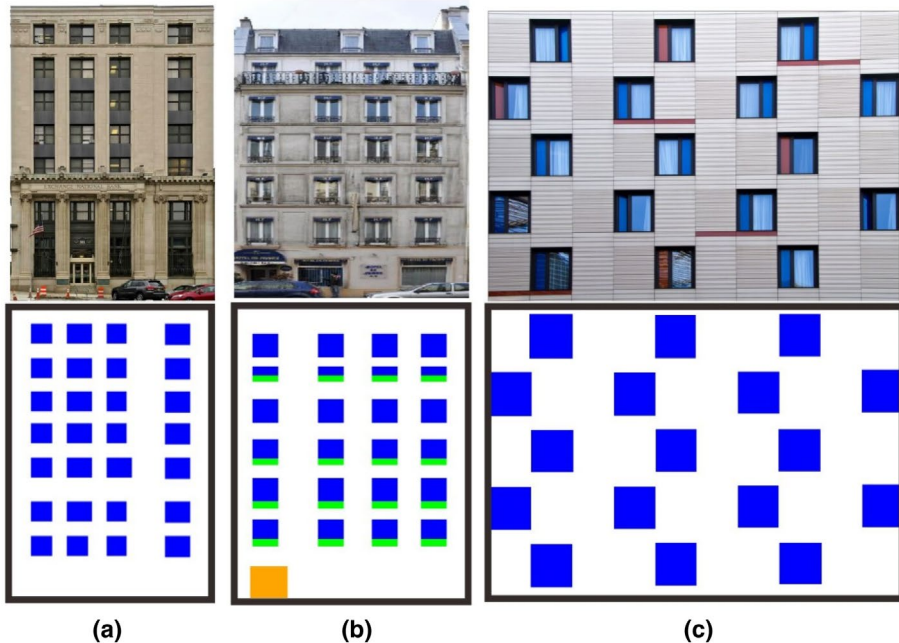


FIGURE 12 Some examples of irregular facades: (a) Bld-5; (b) Bld-6; and (c) Bld-7

to reconstruct using past methods, such as the methods of Bao et al. (2013) and Shen et al. (2016). However, using the proposed second-layer topological graph, each window is accurately reconstructed.

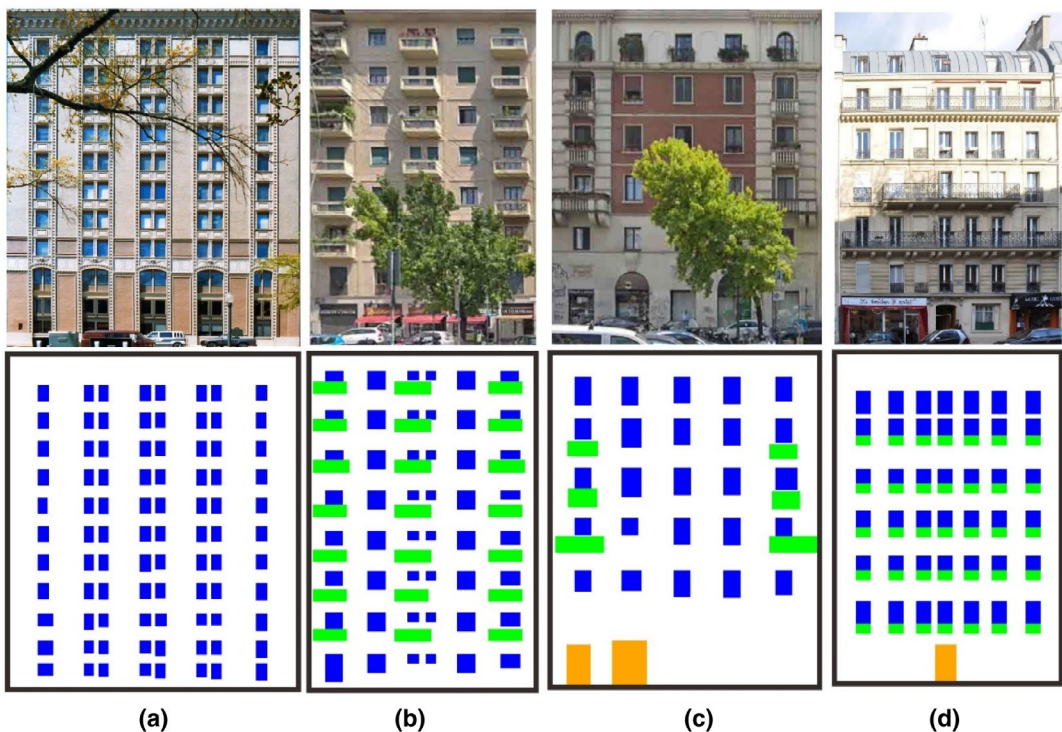
### 5.1.3 | Facades with occlusion

We also use the proposed method to test general facades with different occlusion conditions. Building occlusion on a ground image is different from an aerial image (Zhou, Wang, Tao, Ye, & Wei, 2017). We cannot detect occlusion by constructing the projective geometry. The method proposed in this article can correctly infer the occlusion part on the ground building image. The layouts of the facades are different in Figure 13. In the search algorithm of the first-layer layout graph, we deduce the layouts of the balconies and windows. Therefore, both of these features can be well reconstructed. Second-layer graphs are also applicable to the reconstruction, which enables the facade components to have neat arrangements and constrained topological relationships.

### 5.1.4 | Complex facades

Complex facades often appear because of their aesthetic structure. These facades usually have many windows and complex structures. More importantly, it is difficult to obtain the complete structure with a hand-held camera. As a result, occlusion and deletion always occur. We attempt to use the first-layer layout graph to search for the proper layout of complex facades. The results show that the method still has good performance.

Moreover, we also infer the facade structure by using a grammar-based method. Although the grammar-based method repairs the occluded windows, the time consumption is large. For Figure 14a, due to serious occlusion, the grammar-based method cannot obtain the rules. Hence, the facade cannot be reconstructed.

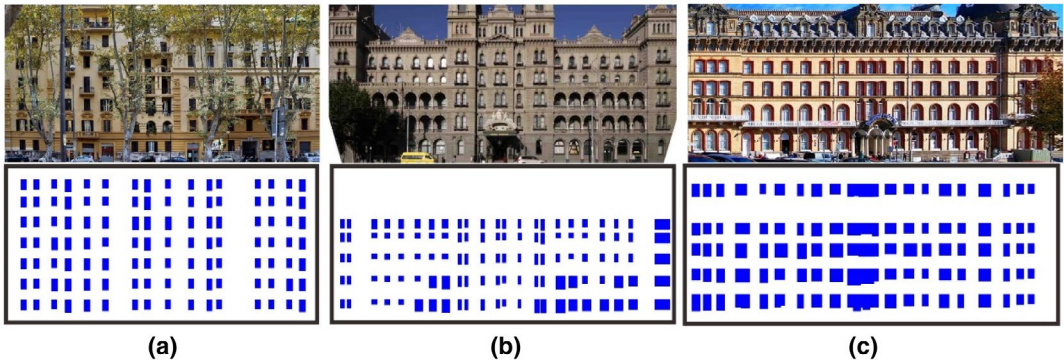


**FIGURE 13** Facades with occlusion: (a) Bld-8; (b) Bld-9; (c) Bld-10; and (d) Bld-11

## 5.2 | Discussion

### 5.2.1 | Performance of the proposed method

The advantages of the proposed method are described in Section 1. From the experimental results, this method can realize parsing and automatic reconstruction of common facades (e.g., Figure 11), irregular facades (e.g., Figure 12), and heavily occluded facades (e.g., Figure 13). Moreover, this method is superior to the previous methods (Teboul et al., 2011) in terms of its accuracy (Table 1) and time performance (Table 2). These superior results are mainly because the parsing of facade images is conducted according to the overall layout of the facades. The essence of the first-layer layout graph is a global optimization process. In this process, we did not actually pay attention to the specific locations of the components, which eliminates the influence of occlusions, shadows, and deletions on the layout inference. The second layer is then mainly used to deal with irregular facades, such as that in Figure 12c. In the constraints we set, we mainly consider frequently occurring phenomena. Therefore, the irregular facades mentioned in this article are still in accordance with the principle of architectural form.



**FIGURE 14** Complex facades from Flickr: (a) Bld-12; (b) Bld-13; and (c) Bld-14

**TABLE 1** Comparison of the accuracy

Method	Type										
	Regular				Irregular		Occlusion		Complex		
	Bld-1	Bld-2	Bld-3	Bld-4	Bld-5	Bld-6	Bld-7	Bld-9	Bld-13	Bld-14	Minimum
Proposed method	1.00	1.00	0.85	0.87	0.94	0.84	1.00	0.83	0.85	0.88	0.83
Grammar-based method	1.00	1.00	0.85	0.90	0.86	0.80		0.7	0.77	0.80	0.7

**TABLE 2** Mean time consumption

	Mean computing time (s)	Step time (s)
Proposed method	6	MBR calculation: 5.4 Applying constraints: 0.2 Type matching and EM: 0.2
Shape grammar-based method	15	Applying constraints: 0.5 Searching grammar: 0.5 Grammar parsing: 14



### Results of the proposed method

First, the facade layout in Figure 15a needs to be deduced using the first-layer layout graph. In the process of deduction there is no proper layout in the first search. Therefore, the proposed algorithm in Section 4.1.3 is run to search for a proper layout. The sequence is changed from  $\{-1, 1, 0\}$  to  $\{0\}$ . The output graph can be described as  $G_1 = \langle V, E \rangle$ , where:

$$V = \left\{ \begin{array}{l} v_1 (\text{window}, 7, 23, 36) \\ v_2 (\text{balcony}, 7, 23, 15) \end{array} \right\} \quad (9)$$

$$E = \{e_1 (v_1, \text{null}) = (d_{\text{low}}, 0)\} \quad (10)$$

From the description above, the established graph is a briefer form for describing the facade than the result of the grammar-based method. The accuracy of the reconstructed facade is shown in Table 1.

### Evaluation

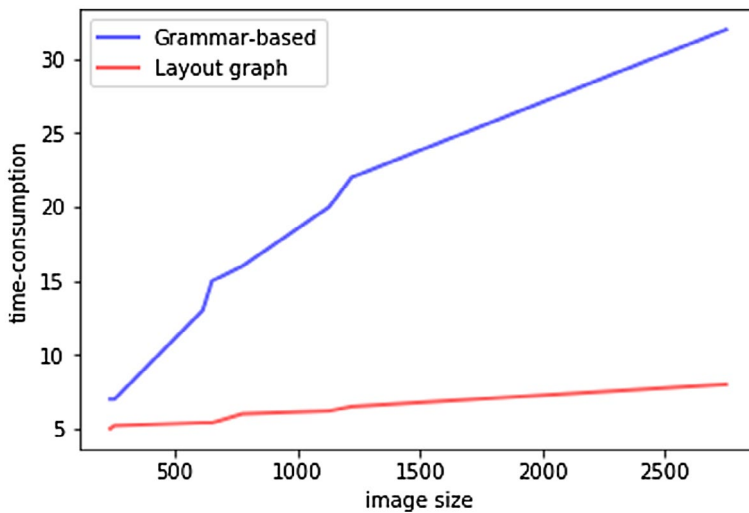
In Table 1, the accuracy of the proposed method is compared to the method of shape grammar based on reinforcement learning. We check the number and location of the reconstructed windows and compare them with the actual facade (the ground-truth constructed through Google Street View). Google Street View can provide multi-angle facade images of the buildings. First, we choose a facade image without occlusion by adjusting the attributes of the angle and shooting time. Then, we make a geometric correction of the facade image, which is consistent with the geometric correction of the VGI image. This ensures correct matching of the positions of the semantic entities in the two images. Finally, the *LabelMe* annotation tool is used to manually annotate the semantic information and generate the ground-truth in order to derive the accuracy.

For different types of facades, we compare the minimum precision values of the two methods (bold numbers in Table 1). This comparison can more objectively reflect the robustness and accuracy of the two methods than the average accuracy. From Table 1, it is observed that the grammar-based method has a strong dependence on the style of facade. For example, it cannot handle the facade of Bld-7 in Figure 12c. Moreover, the grammar-based method relies more on the quality of the facades than our method. This is shown in Figure 15a, which is a facade without occlusion. We can achieve the semantic reconstruction of the facade with the grammar-based method. Using the proposed method, we can also deduce a complete facade model. However, there are some small errors in our results, for example, the magenta windows do not exist in the original facade. This is because the original facade layout is sparse, which does not meet the constraints we set. Therefore, we cannot achieve an accurate facade reconstruction. While we believe adding a row of windows will make the facade more aesthetic, that is not the reality of the situation.

From the time consumption of the two methods (Figure 16, Table 2), the proposed method is more effective than the grammar-based method in terms of global calculations. In terms of a step-by-step time representation, both methods apply their respective constraints. However, the computational time of our constraints is less than the corresponding times of grammar-based methods. In addition, grammar-based methods have more stringent constraints, for example, the arrangement of the windows needs to be fixed. When searching for a proper layout, we are concerned about three abnormal situations (facades with features that are alternating, severely occluded, or inconsistent in size). Therefore, when dealing with abnormal facades, our method can find the optimal solution very quickly. Most importantly, the solution space has been constrained by the topological correlation.

## 5.2.2 | Potential limitations of this method

The proposed method reconstructs a facade structure by verifying the rationality of the facade layout. In the first layer, the facade layout is determined by the control function and the layout sequence. The second layer is



**FIGURE 16** Relationship between image size and time consumption

then used to reconstruct each semantic entity. Although the proposed method can deal with regular, irregular, and complex facades, there are some potential constraints in the process. In the constraints we set, we mainly consider phenomena that occur frequently. Therefore, the irregular facades mentioned in this article are still in accordance with the principle of architectural form. In addition, our method can currently only handle a single facade. In other words, we cannot reconstruct a building image with multiple facades. It should also be noted that the ground-truth we use for evaluation is hand-marked. There may be some small errors, which is inevitable because we cannot obtain the real value.

## 6 | CONCLUSIONS

A facade is a type of man-made object that has a regular arrangement. In this article, we explored the geometric and topological consistencies in the arrangement of facade components. We proposed a novel method to parse facade images and reconstruct facades. According to the principle of architectural form, the overall layout of a facade is constrained by the control function. Thus, we can deduce and reconstruct a complete facade according to the hierarchical layout graph. The proposed method improved some of the problems with traditional methods of facade parsing:

1. Traditional methods have poor robustness due to the influence of the architectural style and image size when using an inferred grammar. Because the proposed method analyses the facade from the overall layout, it is not sensitive to noise, occlusions, or shadows.
2. The size of the image cannot influence the calculation time when deducing a layout graph, as we set a reasonable threshold to restrict the spacing distance between components.
3. This method has a strong adaptability when inferring the different styles and complex facades.

In addition, the use of a topological graph makes storing the layout features of the facade easier. This benefit will help us build a large-scale database of the building facade models. In future work, we will use OpenStreetMap to store the building facade information in the covered area. Furthermore, the facade layout graph can be translated into CityGML form (Gröger & Plümer, 2012), which can help to achieve a large-scale 3D city model.

## ORCID

Hongchao Fan  <https://orcid.org/0000-0002-0051-7451>

## NOTE

<sup>1</sup>A Haussmannian building is a kind of typical Paris architecture. Teboul et al. (2011) have constructed a dataset of ECP2011 which consists of 104 annotated images of Haussmannian buildings in Paris.

## REFERENCES

- Alegre, F., & Dellaert, F. (2004). A probabilistic approach to the semantic interpretation of building facades. In *Proceedings of the International Workshop on Vision Techniques Applied to the Rehabilitation of City Centres*. Lisbon, Portugal: CIPA.
- Bao, F., Schwarz, M., & Wonka, P. (2013). Procedural facade variations from a single layout. *ACM Transactions on Graphics*, 32(1), 1–13.
- Becker, S. (2009). Generation and application of rules for quality dependent facade reconstruction. *ISPRS Journal of Photogrammetry & Remote Sensing*, 64(6), 640–653.
- Berg, A. C., Grabler, F., & Malik, J. (2007). Parsing images of architectural scenes. In *Proceedings of the 11th IEEE International Conference on Computer Vision*, Rio de Janeiro, Brazil (pp. 1–8). Piscataway, NJ: IEEE.
- Cheng, M. M., Zhang, G. X., Mitra, N. J., Huang, X., & Hu, S. M. (2011). Global contrast based salient region detection. In *Proceedings of the 24th IEEE Conference on Computer Vision and Pattern Recognition*, Colorado Springs, CO (pp. 409–416). Piscataway, NJ: IEEE.
- Ching, F. D. (2014). *Architecture: Form, space, and order*. New York, NY: John Wiley & Sons.
- Cohen, A., Schwing, A., & Pollefeys, M. (2014). Efficient structured parsing of facades using dynamic programming. In *Proceedings of the 27th IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, OH (pp. 3206–3213). Piscataway, NJ: IEEE.
- Dai, D., Prasad, M., Schmitt, G., & Van Gool, L. (2012). Learning domain knowledge for facade labelling. In A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, & C. Schmid (Eds.), *Computer vision: ECCV 2012* (Lecture Notes in Computer Science, Vol. 7578, pp. 710–723). Berlin, Germany: Springer.
- Datta, R., Joshi, D., Jia, L., & Wang, J. Z. (2008). Image retrieval: Ideas, influences, and trends of the new age. *ACM Computing Surveys*, 40(2), 1–60.
- Doersch, C., Singh, S., Gupta, A., Sivic, J., & Efros, A. A. (2012). What makes Paris look like Paris? *ACM Transactions on Graphics (SIGGRAPH)*, 31(4), 1–9.
- European Parliament. (2018). Directive on the energy performance of buildings (2018/844/EU). *The Official Journal of the European Union*, L156, 75–91.
- Fathalla, R., & Vogiatzis, G. (2016). A deep learning pipeline for semantic facade segmentation. In *Proceedings of the British Machine Vision Conference*. York, UK: BMVA.
- Felzenszwalb, P. F., & Huttenlocher, D. P. (2004). Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2), 167–181.
- Flemming, U. (1990). Syntactic structures in architecture: Teaching composition with computer assistance. In M. McCullough, W. J. Mitchell, & P. Purcell (Eds.), *The electronic design studio: Architectural knowledge and media in the computer era* (pp. 31–48). Cambridge, MA: MIT Press.
- Gadde, R., Jampani, V., Marlet, R., & Gehler, P. (2018). Efficient 2D and 3D facade segmentation using auto-context. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 40(5), 1273–1280.
- Gadde, R., Marlet, R., & Paragios, N. (2016). Learning grammars for architecture-specific façade parsing. *International Journal of Computer Vision*, 117(3), 290–316.
- Gröger, G., & Plümer, L. (2012). CityGML-interoperable semantic 3D city models. *ISPRS Journal of Photogrammetry & Remote Sensing*, 71, 12–33.
- Haala, N., Rothermel, M., & Cavegn, S. (2015). Extracting 3D urban models from oblique aerial images. In *Proceedings of the Joint Urban Remote Sensing Event*. Lausanne, Switzerland: IEEE.
- Hsu, Y. C. (2004). Space adjacency behavior in space planning. In *Proceedings of the Ninth Conference on Computer-Aided Architectural Design Research in Asia*. Seoul, South Korea: SAGE.
- Hu, S. M., Zhang, F. L., Wang, M., Martin, R. R., & Wang, J. (2013). PatchNet: A patch-based image representation for interactive library-driven image editing. *ACM Transactions on Graphics*, 32(6), 1–12.
- Jampani, V., Gadde, R., & Gehler, P. V. (2015). Efficient facade segmentation using auto-context. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision* (pp. 1038–1045). Waikoloa Beach, Hawaii: IEEE.
- Jennath, K. A., & Nidhish, P. J. (2016). Aesthetic judgement and visual impact of architectural forms: A study of library buildings. *Procedia Technology*, 24, 1808–1818.

- Koutsourakis, P., Simon, L., Teboul, O., Tziritas, G., & Paragios, N. (2009). Single view reconstruction using shape grammars for urban environments. In *Proceedings of the 12th IEEE International Conference on Computer Vision* (pp. 1795–1802). Kyoto, Japan: IEEE.
- Krier, R., & Vorreiter, G. (1988). *Architectural composition*. New York, NY: Rizzoli.
- Ladicky, L. U., Russell, C., Kohli, P., & Torr, P. H. S. (2009). Associative hierarchical CRFs for object class image segmentation. In *Proceedings of the 12th IEEE International Conference on Computer Vision* (pp. 739–746). Kyoto, Japan: IEEE.
- Lee, S. C., & Nevatia, R. (2004). Extraction and integration of window in a 3D building model from ground view images. In *Proceedings of the 2004 IEEE International Conference on Computer Vision and Pattern Recognition*, Washington, DC (pp. 113–120). Piscataway, NJ: IEEE.
- Li, D. H. W., & Lam, J. C. (2001). Evaluation of lighting performance in office buildings with daylighting controls. *Energy & Buildings*, 33(8), 793–803.
- Liu, H., Zhang, J., Zhu, J., & Hoi, S. C. (2017). DeepFacade: A deep learning approach to facade parsing. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence* (pp. 2301–2307). Melbourne, Australia: IJCAI.
- Lotte, R., Haala, N., Karpina, M., Aragão, L., & Shimabukuro, Y. (2018). 3D Facade labeling over complex scenarios: A case study using convolutional neural network and structure-from-motion. *Remote Sensing*, 10(9), 1435.
- Martinović, A., Mathias, M., Weissenberg, J., & Gool, L. V. (2012). A three-layered approach to facade parsing. In A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, & C. Schmid (Eds.), *Computer vision: ECCV 2012* (Lecture Notes in Computer Science, Vol. 7578, pp. 416–429). Berlin, Germany: Springer.
- Mathias, M., Martinović, A., & Van Gool, L. (2016). ATLAS: A three-layered approach to facade parsing. *International Journal of Computer Vision*, 118(1), 22–48.
- Müller, P., Gang, Z., Wonka, P., & Gool, L. J. V. (2007). Image-based procedural modeling of facades. *ACM Transactions on Graphics*, 26(3), 85.
- Müller, P., Wonka, P., Haegler, S., Ulmer, A., & Van Gool, L. (2006). Procedural modeling of buildings. *ACM Transactions on Graphics*, 25(3), 614–623.
- Pu, S., & Vosselman, G. (2009). Building facade reconstruction by fusing terrestrial laser points and images. *Sensors*, 9(6), 4525–4542.
- Riemenschneider, H., Krispel, U., Thaller, W., Donoser, M., Havemann, S., Fellner, D., & Bischof, H. (2012). Irregular lattices for complex shape grammar facade parsing. In *Proceedings of 2012 IEEE Conference on Computer Vision and Pattern Recognition, Prudence, Rhode Island* (pp. 1640–1647). Piscataway, NJ: IEEE.
- Salvan, G. S., & Thapa, S. (2000). *Architectural & construction data: A digested book for daily use*. Quezon City, Philippines: Goodwill Bookstore.
- Schmitz, M., & Mayer, H. (2016). A convolutional network for semantic façade segmentation and interpretation. *ISPRS International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences*, XLI-B3, 709–715.
- Schwartz, B. L., & Krantz, J. H. (2017). *Sensation and perception*. Thousand Oaks, CA: Sage Publications.
- Shen, J., Fan, H., Mao, B., & Wang, M. (2016). Typification for façade structures based on user perception. *ISPRS International Journal of Geo-Information*, 5(12), 239.
- Stamps, A. E. III. (1999). Physical determinants of preferences for residential facades. *Environment & Behavior*, 31(6), 723–751.
- Teboul, O., Kokkinos, I., Simon, L., Koutsourakis, P., & Paragios, N. (2011). Shape grammar parsing via reinforcement learning. In *Proceedings of the 24th IEEE Conference on Computer Vision and Pattern Recognition*, Colorado Springs, CO (pp. 2273–2280). Piscataway, NJ: IEEE.
- Teboul, O., Simon, L., Koutsourakis, P., & Paragios, N. (2013). Segmentation of building facades using procedural shape priors. In *Proceedings of the 2013 IEEE International Conference on Computer Vision & Pattern Recognition*, San Francisco, CA (pp. 3105–3112). Piscataway, NJ: IEEE.
- Tyleček, R. (2012). *The CMP facade database* (Research Report CTU-CMP-2012-24). Prague, Czech Republic: Czech Technical University.
- Tyleček, R., & Šára, R. (2010). A weak structure model for regular pattern recognition applied to facade images. In *Proceedings of Asian Conference on Computer Vision* (pp. 450–463). Berlin, Heidelberg: Springer.
- Uden, M., & Zipf, A. (2013). Open building models: Towards a platform for crowdsourcing virtual 3D cities. In J. Pouliot, S. Daniel, F. Hubert, & A. Zamyadi (Eds.), *Progress and new trends in 3D geoinformation sciences* (Lecture Notes in Geoinformation & Cartography, pp. 299–314). Berlin, Germany: Springer.
- Wang, Y., Ma, Y., Zhu, A. X., Zhao, H., & Liao, L. (2018). Accurate facade feature extraction method for buildings from three-dimensional point cloud data considering structural information. *ISPRS Journal of Photogrammetry & Remote Sensing*, 139, 146–153.
- Wenzel, S., & Förstner, W. (2012). Learning a compositional representation for facade object categorization. *ISPRS Annals of Photogrammetry, Remote Sensing & the Spatial Information Sciences*, 1–3, 197–202.
- Xiao, J., Fang, T., Tan, P., Zhao, P., Ofek, E., & Quan, L. (2008). Image-based facade modeling. *ACM Transactions on Graphics*, 27(5), 161.



- Yang, M. Y., & Förstner, W. (2011). A hierarchical conditional random field model for labeling and classifying images of man-made scenes. In *Proceedings of the 2011 IEEE International Conference on Computer Vision Workshops*, Barcelona, Spain (pp. 196–203). Piscataway, NJ: IEEE.
- Zhang, L., & Liang, Z. (2017). Deep learning-based classification and reconstruction of residential scenes from large-scale point clouds. *IEEE Transactions on Geoscience & Remote Sensing*, 99, 1–11.
- Zhou, G., Luo, Q., Xie, W., Tao, Y., Huang, J., & Shen, Y. (2016). Transformation model with constraints for high-accuracy of 2D–3D building registration in aerial imagery. *Remote Sensing*, 8(6), 507.
- Zhou, G., Wang, Y., Tao, Y., Ye, S., & Wei, W. (2017). Building occlusion detection from ghost images. *IEEE Transactions on Geoscience & Remote Sensing*, 55(2), 1074–1084.

**How to cite this article:** Wang Y, Fan H, Zhou G. Reconstructing facade semantic models using hierarchical topological graphs. *Transactions in GIS*. 2020;24:1073–1097. <https://doi.org/10.1111/tgis.12616>